

## Appendix B

### 2.5. Class Definition

```

class RuleGenerator{
private:
    Stack    m_redoStack, m_undoStack;
    Page     *m_pFirstPage, *m_pLastPage;
    Chain    *m_pDelChain;
    Chain    *m_pFirstChain, *m_pLastChain;
    Chain    *m_pFirstDelChain, *m_pLastDelChain;
    bool     m_bCanUndo, m_bCanRedo;
public:
    PageRule m_FirstRule;
private:
    void DestroyElement(Element *pElement);
    Element *DeleteElement(Chain *pChain, Element *pElement);
    void LocateElement(Chain *pChain, ElementInfo *pElementInfo);
    void NewChain(ElementInfo *pElementInfo, Statement *pStatement);
    Chain *DeleteChain(Chain *pChain, Card *pCard=NULL, int
mode=FROM_NOWHERE);
    void NewChildChain(Element *pParentElement, char cAction, Element
*pSourceElement);
    void UpdateChain(ElementInfo *pElementInfo, Statement *pStatement);
    void UpdateDelChain(TagId Ele, char cAction);
    void FilterDelChain();
    void AssemblyChain();
    void ParseFrame(Chain *pChain, Page *pPage);
    void OutputVar(ostream& xsltFile, Var *pVar, DOMString sFrom);
    void OutputChain(ostream& xsltFile, Card *pCard, Chain *pChain);
    Element *GetUnit(Card *pCard);
public:
    RuleGenerate ();           //initialize the redo & undo stack(alloc memory for them)
    ~RuleGenerate();          //free the space
public:
    void PushStatement(char cAction, TagId sourceEle, TagId targetEle, bool bNewAction);
    void PushStatement(char cAction, TagId sourceEle, bool bNewAction);
    void PushStatement(TagId sourceEle, DOMString psNewAttrName[], DOMString
psNewAttrValue[], int iNumOfAttr, bool bNewAction);
    void PushStatement(TagId sourceEle, DOMString sNewText, bool bNewAction)
    void UndoStatement();
    void RedoStatement();
    bool CanRedo();
    bool CanUndo();
    void GenerateXSLT();
    char *SaveTempXSLT(int iPage, int iFrame);
    int ReloadXSLT(char *pszXsltFile, int iPage, int iFrame);
};

```